

Course: Database Management System

Real Estate Broker System

Bastawala Asad Mohd. Idris	(202012037)
Maruf Memon	(202012038)
Kashish Rajeshbhai Shah	(202012039)
Sanghvi Vruddhi Pareshbhai	
(202012040)	Nirmit Gupta
(202018019)	K Prathyusha
(202018020)	

10/09/2020

Queries:

1. By using the filters like city code of any city, buyers will be able to view all the properties available for sale in the city.

Relational Algebra

$$r1 = (\pi_{(city_id)} (\sigma_{(city_name='Gandhinagar')} (city)))$$

$$result = \sigma_{(city_id=r1)} (property)$$

SQL Query

```
select * from property where city_id in (select city_id from city where city_name='Gandhinagar');
```

Query Editor

Query History

Scratch P.

```

1 set search_path to realestate;
2 select * from property where city_id in (select city_id from city where city_name='Gandhinagar');
3

```

Data Output

Explain

Notifications

Messages

	property_id [PK] integer		property_name character varying (20)		price real		features realestate.features_t		property_type realestate.property_t		amenities realestate.property_a		property_layout character varying (10)		property_desc character varying (200)		address character varying (100)
1	1	Vrundavan Flats	5e+06	Standard	Residential	Furnished	2500	Spacious	B-10 Vr								
2	2	3rd eye vision	9e+06	Premium	Commercial	Unfurnished	5000	Full Security	9 Torqu								

2. Buyer can find the seller name and contact of a particular property by searching by property name.

Relational Algebra

$$r1 = (\pi_{(seller_id)} (\sigma_{(property_name='Vrundavan Flats')} (property)))$$

$$result = (\pi_{(name, contact_no)} (\sigma_{(user_id=r1)} (person \bowtie_{(person.user_id=person_contact.user_id)} person_contact)))$$

SQL Query

```
select u.name, pc.contact_no from person as u JOIN person_contact as pc on u.user_id=pc.user_id where u.user_id in (select seller_id from property where property_name='Vrundavan Flats');
```

property_name='Vrundavan Flats');



The screenshot shows a PostgreSQL query editor interface. At the top, the title bar reads 'postgres/postgres@PostgreSQL 12'. Below it are tabs for 'Query Editor' and 'Query History'. The query editor contains the following SQL code:

```
1 set search_path to realestate;
2 select u.name,pc.contact_no from person as u JOIN person_contact as pc on u.user_id=pc.user_id
3 where u.user_id in (select seller_id from property where property_name='Vrundavan Flats');
```

Below the query editor are tabs for 'Data Output', 'Explain', 'Notifications', and 'Messages'. The 'Data Output' tab is active, displaying the results of the query in a table format:

	name character varying (50)	contact_no numeric (10)
1	Maria Mason	8989894482

3. Buyer can find the list of all properties and corresponding seller names,contact nos that fall under a particular category(feature) of property that he is interested in buying.

Relational Algebra

$(\pi_{(name,contact_no,property_*)} (\sigma_{(features='Premium')} (person \bowtie_{(person.user_id=property.seller_id)} property \bowtie_{(person.user_id=person_contact.user_id)} person_contact)))$

SQL Query

```
select u.name,pc.contact_no,p.* from person as u JOIN property as p ON u.user_id = p.seller_id JOIN person_contact as pc on u.user_id=pc.user_id where p.features='Premium';
```

postgres/postgres@PostgreSQL 12										
Query Editor Query History										
<pre> 1 set search_path to realestate; 2 select u.name,pc.contact_no,p.* from person as u JOIN property as p ON u.user_id = p.seller_id 3 JOIN person_contact as pc on u.user_id=pc.user_id where p.features='Premium'; </pre>										
Data Output Explain Notifications Messages										
	name character varying (50)	contact_no numeric (10)	property_id integer	property_name character varying (20)	price real	features realestate.features_t	property_type realestate.property_t	amenities realestate.property_a	property_layout character varying (1	
1	Charles Gonzalez	8989894481	3	Pentagon Complex	7e+06	Premium	Commercial	Furnished	2000	

4. Display names of all the buyers who are going to visit property 'Pentagon Complex'

Relational Algebra

$$(\pi_{(name)} (\sigma_{(property_name='Pentagon\ Complex')} (person \bowtie_{(person.user_id=visit.buyer_id)} visit \bowtie_{(property.property_id=visit.property_id)} property)))$$

SQL Query

```

select name from person as p JOIN visit ON p.user_id = visit.buyer_id join property as pr
on visit.property_id=pr.property_id where pr.property_name = 'Pentagon Complex';

```

postgres/postgres@PostgreSQL 12

Query Editor Query History

```

1 select name from person as p JOIN visit ON p.user_id = visit.buyer_id join property as pr
2 on visit.property_id=pr.property_id where pr.property_name = 'Pentagon Complex';
3 |

```

Data Output Explain Notifications Messages

	name
1	Doris Elliott

5. Display details of all the properties in Gandhinagar city whose price is between 5000000 and 7000000

Relational Algebra

$$r1 = (\pi_{(city_id)} (\sigma_{(city_name='Gandhinagar')} (city)))$$

$$result = \sigma_{(city_id=r1 \text{ and } price > 5000000 \text{ and } price < 7000000)} (property)$$

SQL Query

```
select * from property where city_id in (select city_id from city where city_name='Gandhinagar') and (price >= 5000000 and price < 7000000);
```

postgres/postgres@PostgreSQL 12										
Query Editor Query History										
<pre> 1 set search_path to realstate; 2 select * from property where city_id in (select city_id from city where city_name='Gandhinagar') and (price>=5000000 and price<7000000); </pre>										
Data Output Explain Notifications Messages										
	property_id [PK] integer	property_name character varying (20)	price real	features realstate.features_t	property_type realstate.property_t	amenities realstate.property_a	property_layout character varying (10)	property_desc character varying (200)	address character varying (255)	
1	1	Vrundavan Flats	5e+06	Standard	Residential	Furnished	2500	Spacious	B-10 Vr	

6. Display the details of seller who sold the property to buyer with seller 'Maria Mason' through agent 'William Mendoza'

Relational Algebra

$$r1 = (\pi_{(user_id)} (\sigma_{(person.name='Maria\ Mason')} (person)))$$

$$r2 = (\pi_{(user_id)} (\sigma_{(person.name='William\ Mendoza')} (person)))$$

$$r3 = (\pi_{(user_id)} (\sigma_{(seller_id=r1\ and\ agent_id=r2)} (person \bowtie_{(person.user_id=property.seller_id)} property))))$$

$$result = \sigma_{(user_id=r3)} (person \bowtie_{(person.user_id=person_contact.user_id)} person_contact)$$

SQL Query

```

select * from person as u join person_contact as pc on u.user_id=pc.user_id
where u.user_id in

```

```

(select person.user_id from person JOIN property ON person.user_id =
property.seller_id WHERE

```

```

property.seller_id = (select person.user_id from person where
person.name='Maria Mason') AND

```

property.agent_id = (select person.user_id from person where person.name='William Mendoza'));

postgres/postgres@PostgreSQL 12

Query Editor Query History

```

1 select * from person as u join person_contact as pc on u.user_id=pc.user_id where u.user_id in
2 (select person.user_id from person JOIN property ON person.user_id = property.seller_id WHERE
3 property.seller_id = (select person.user_id from person where person.name='Maria Mason') AND
4 property.agent_id = (select person.user_id from person where person.name='William Mendoza'));

```

Data Output Explain Notifications Messages

	user_id integer	name character varying (50)	address character varying (100)	dob date	gender character (1)	user_id integer	contact_no numeric (10)
1	2	Maria Mason	30 Memorial Drive, Avon MA 2...	1973-07...	F	2	8989894482

7. Display details to get all the bookings made by a particular buyer.

Relational Algebra

$r1 = (\pi_{(user_id)} (\sigma_{(name='Patricia Peterson')} (person)))$

$result = (\sigma_{(buyer_id=r1)} (booking))$

SQL Query

select * from booking where buyer_id in (select user_id from person where name='Patricia Peterson');

postgres/postgres@PostgreSQL 12							
Query Editor Query History							
<pre> 1 set search_path to realestate; 2 select * from booking where buyer_id in (select user_id from person where name='Patricia Peterson'); </pre>							
Data Output Explain Notifications Messages							
	booking_id [PK] integer	booking_status realestate.booking_s	booking_date date	transaction_id integer	buyer_id integer	property_id integer	
1	1	Success	2020-10-10		1	9	1

8. Display The details of property and name of buyer who have booked the visit on 12/09/2019

Relational Algebra

$$r1 = \sigma_{(visit_date='12/09/2019')} (property \bowtie_{(person.property_id = visit.property_id)} visit \bowtie_{(person.user_id = visit.buyer_id)} person))$$

$$result = (\pi_{(name, property.*)} (r1))$$

SQL Query

```
select u.name,p.* from property as p JOIN visit as v ON p.property_id = v.property_id
JOIN person as u ON u.user_id = v.buyer_id where v.visit_date = '12/09/2019';
```


postgres/postgres@PostgreSQL 12

Query Editor Query History

```

1 set search_path to realstate;
2 select u.name,p.* from property as p JOIN visit as v ON p.property_id = v.property_id JOIN person as u ON u.user_id =
3 where v.visit_date = '12/09/2019';

```

Data Output Explain Notifications Messages

	name character varying (50)	property_id integer	property_name character varying (20)	price real	features realstate.features_t	property_type realstate.property_t	amenities realstate.property_a
1	Doris Elliott	3	Pentagon Complex	7e+06	Premium	Commercial	Furnished

9. Display the details of contractor whose price quote is between 40 and 60.

Relational Algebra

result = $(\sigma_{(\text{price_quote} > '40' \text{ and } \text{price_quote} < '60')} (\text{person} \bowtie_{(\text{person.user_id} = \text{contractor.user_id})} \text{contractor}))$

SQL Query

```

select * from Person as u JOIN Contractor as c ON u.user_id = c.user_id WHERE
c.price_quote > '40' and c.price_quote < '60';

```

postgres/postgres@PostgreSQL 12									
Query Editor Query History									
<pre> 1 set search_path to realestate; 2 select * from Person as u JOIN Contractor as c ON u.user_id = c.user_id WHERE c.price_quote > '40' 3 and c.price_quote < '60'; </pre>									
Data Output Explain Notifications Messages									
	user_id integer	name character varying (50)	address character varying (100)	dob date	gender character (1)	price_quote real	user_id integer		
1	25	Steven Romero	13858 Rt 31 W, Albion NY 144...	1963-09...	M	55	25		
2	27	Jerry Turner	101 Sanford Farm Shpg Cente...	1991-05...	M	53	27		
3	31	Norma Dunn	5399 W Genesee St, Camillus ...	1967-07...	F	43	31		
4	70	William George	425 Route 31, Macedon NY 1...	1985-08...	M	46	70		

10. Display the details of all properties sold by agent 'Sharon Schmidt' along with his name.

Relational Algebra

$$r1 = \pi_{(user_id)} (\sigma_{(name='Sharon Schmidt')} (person \bowtie property))$$

$$r2 = (\sigma_{(agent_id=r1)} (person \bowtie property))$$

$$result = (\pi_{(name, property.*)} (r2))$$

SQL Query

Select name,property.* from Property JOIN person ON property.agent_id = person.user_id where

property.agent_id = (select person.user_id from person where person.name='William Mendoza');

```
1 Select name,property.* from Property JOIN person ON property.agent_id = person.user_id where
2 property.agent_id = (select person.user_id from person where person.name='William Mendoza');
```

	<div>name</div> <div>character varying (50)</div>	<div>property_id</div> <div>integer</div>	<div>property_name</div> <div>character varying (20)</div>	<div>price</div> <div>real</div>	<div>features</div> <div>realestate.features_t</div>	<div>property_type</div> <div>realestate.property_t</div>	<div>ammenities</div> <div>realestate.property_a</div>
1	Sharon Schmidt	3	Pentagon Complex	7e+06	Premium	Commercial	Furnished
2	Sharon Schmidt	11	Lounge Bungalows	2e+07	Standard	Residential	Furnished